“New” AI in terms of Agent-Space architecture

RNDr. Andrej Lúčny

MicroStep-MIS & DAI FMFI UK

andy@microstep-mis.com

http://www.microstep-mis.com/~andy

“New” AI

- unlike GOFAI (which focus on universal mechanisms of logical derivation) “New” AI prefers to employ various specific (ad-hoc) mechanisms and look for mechanisms how to put them together to generate adequate global behavior.

Postulates of “New” AI

- (Principle of situated system)
- Embodiment
- Emergence
- Interaction
- Hierarchy

Situated system

- It is designed to not operate under general conditions but under a set of specific conditions
- Former versions can work for a smaller set, later (modified) versions for bigger set
- We do not build one universal handler of general outside conditions; various states of environment have to be handled by various different modules specialized to particular cases. Thus once we succeed thanks to one module, some other time due to activity of another one

Embodiment

- We do not split design from implementation
- We design having a prototype in hand
- After each increment or modification we have possibility to test on the prototype what he have exactly done and we have chance to correct it until all required tests are passed

Emergence

- System behavior is a result of interaction among its units
- Sometimes relationship between the two can be hidden to developer; he is not able to predict it, but he is able to test it and then find its explanation

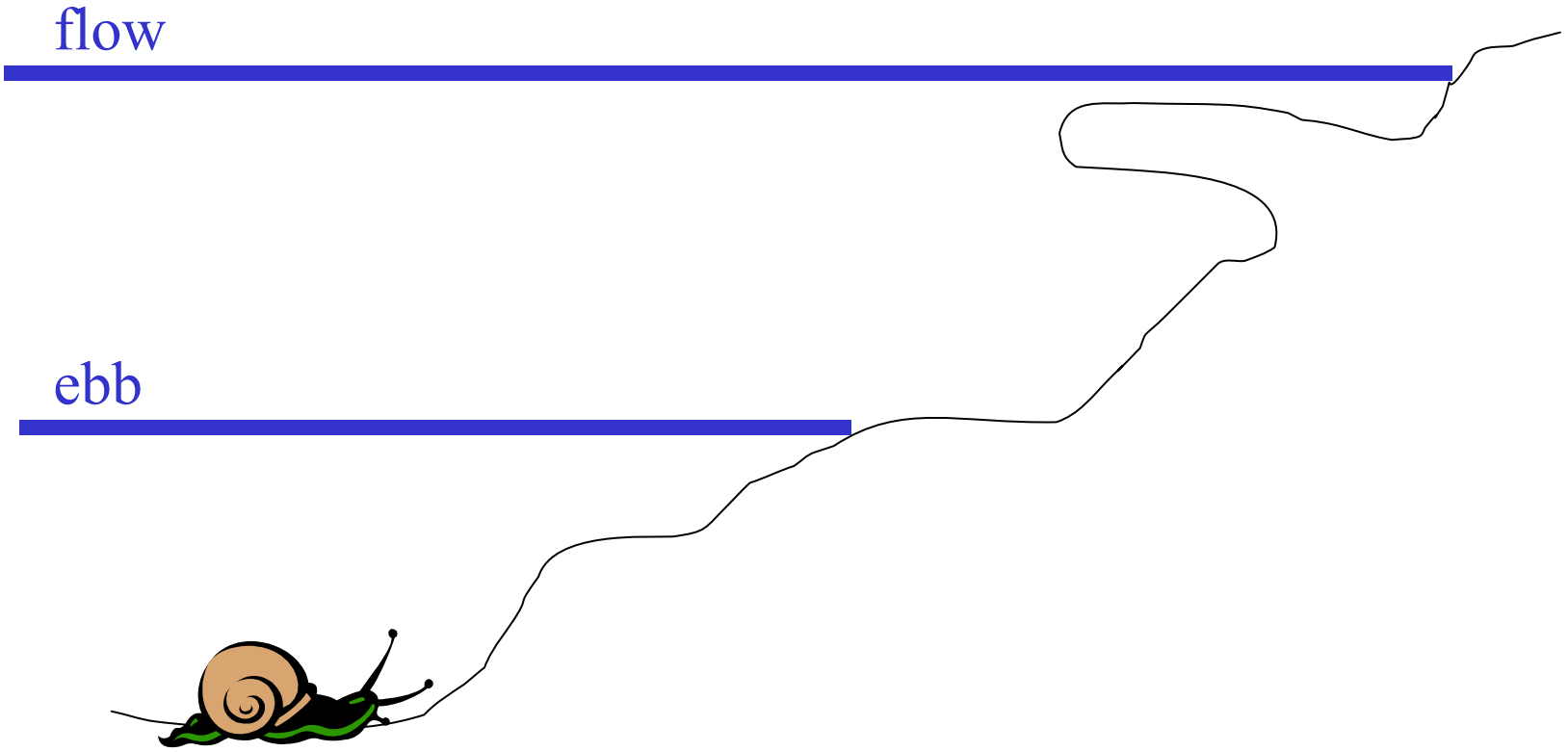
Interaction

Intelligence can emerge on system which has no cognitive modules due to

- interaction among its units
- its interaction with dynamic environment

Exhausting of intelligence from environment

- snail Litorino (dies in aquarium, survives in sea)



Hierarchy

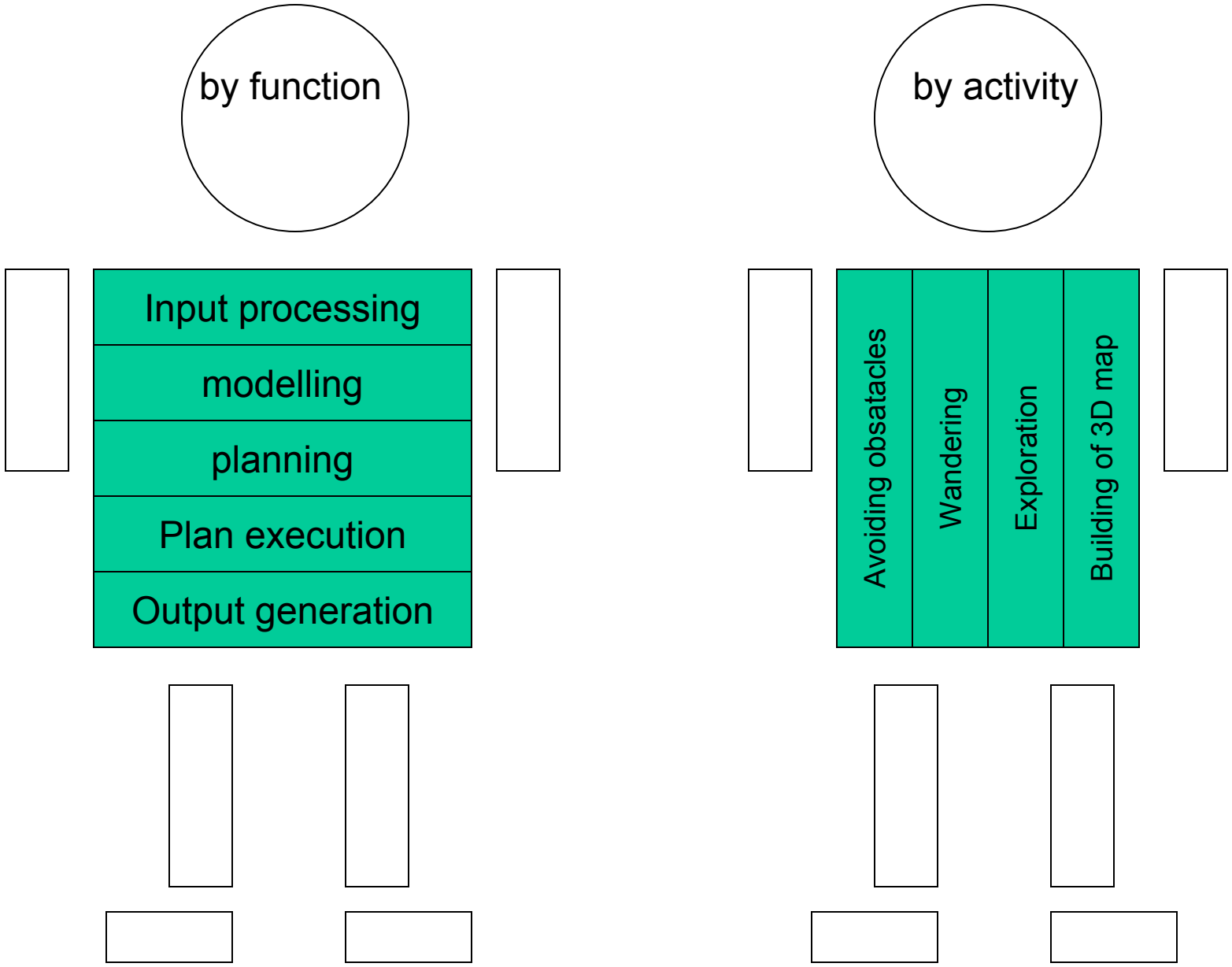
- System is developed incrementally from bottom to up
- We start with simpler layers and sequentially add more difficult ones (biomimethic methaphor)
- Risk: we can get to situation that we are no able to provide any further reasonable increment
- Profit: integration of implemented parts cannot fail
- The later levels can employ and affect the former ones

Decomposition

- By function – to layers which collect modules having similar code, layers create a pipeline

- By activity – to layers which collect modules which are responsible for a particular ingredient of global behavior, layers operate in parallel

  (preferred)


by function
Input processing
modelling
planning
Plan execution
Output generation
by activity
Avoiding obsatacles
Wandering
Exploration
Building of 3D map

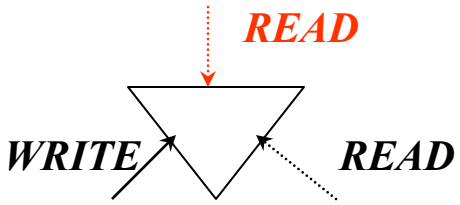
Subsumption

How can a later layer employ and affect a former layer ?

- wire-taping
- inhibition
- suppression

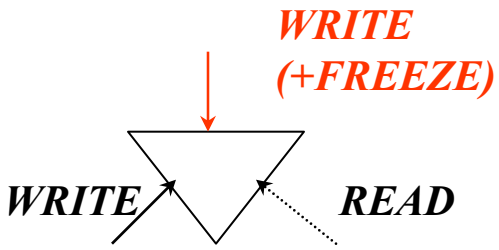
Wiretaping

- A later layer undertakes data which flow between modules in a former layer



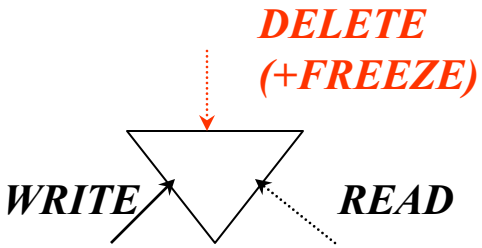
Suppression

- A later layer overwrites data which flow between modules in a former layer



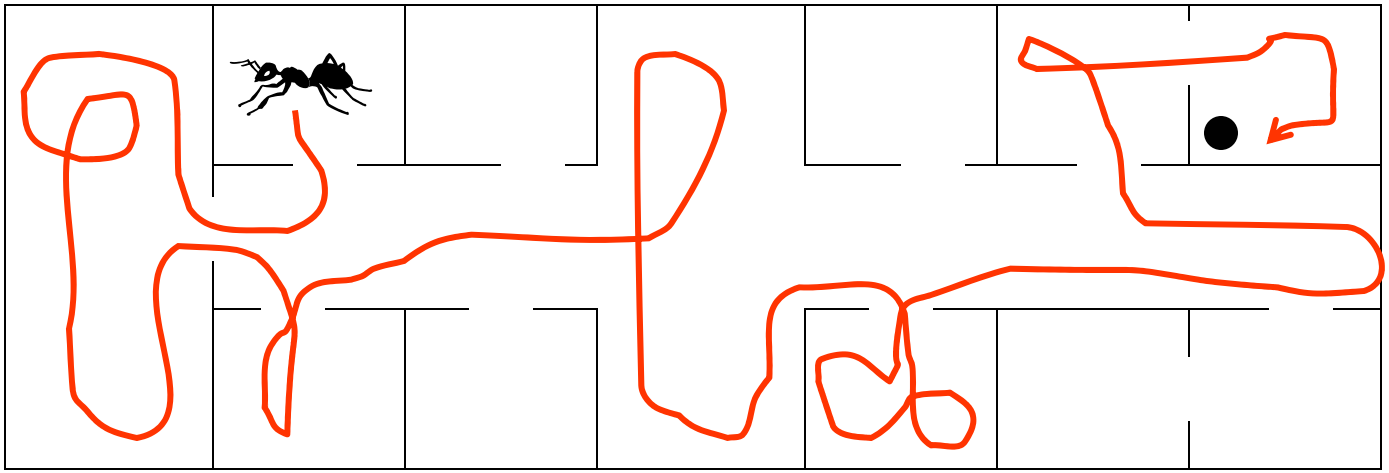
Inhibition

- A later layer erases data and thus stops flow between modules in a former layer

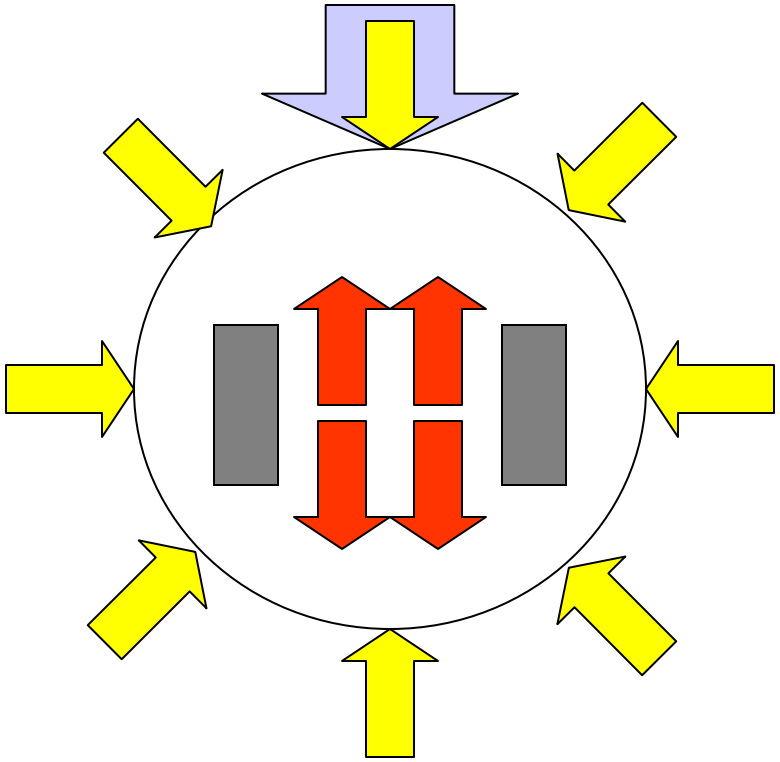


An example

Control system of mobile robot which has to find a ball in environment with walls, doors, passages, furniture …



Sensors and actuators



detection of the closest obstacle

Forward/Backward movement and left/right rotation

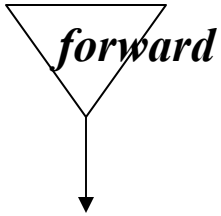
Detection of ball

Design of activities

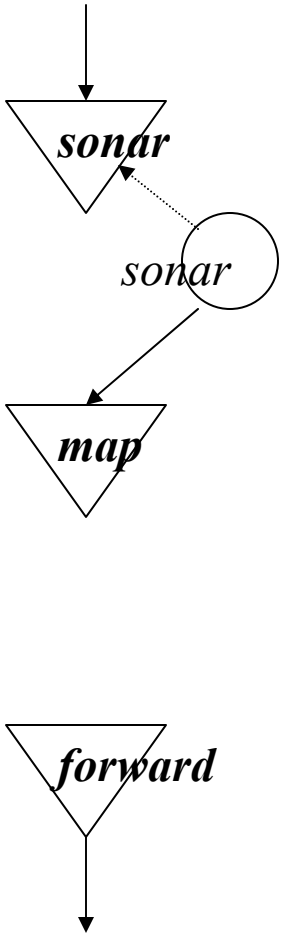
STOP	**– Stop at ball**
EXPLORE	**– Exploration of space**
WANDER	**– Random wandering**
AVOID	**– Avoiding obstacles**

AVOID

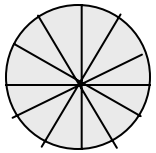
Anytime we try to go forward



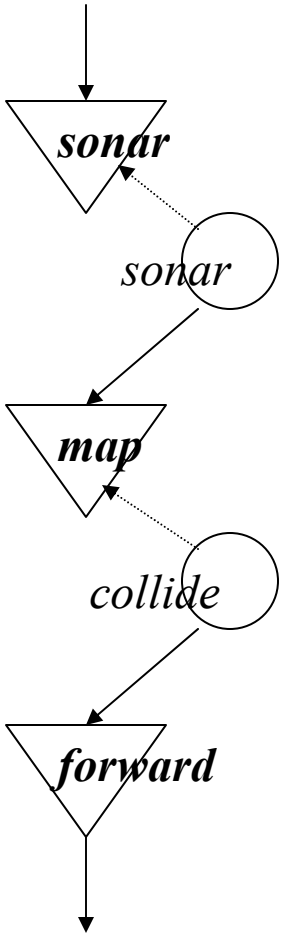
AVOID



Sonar creates a map of distances to the nearest obstacles in individual sections of space



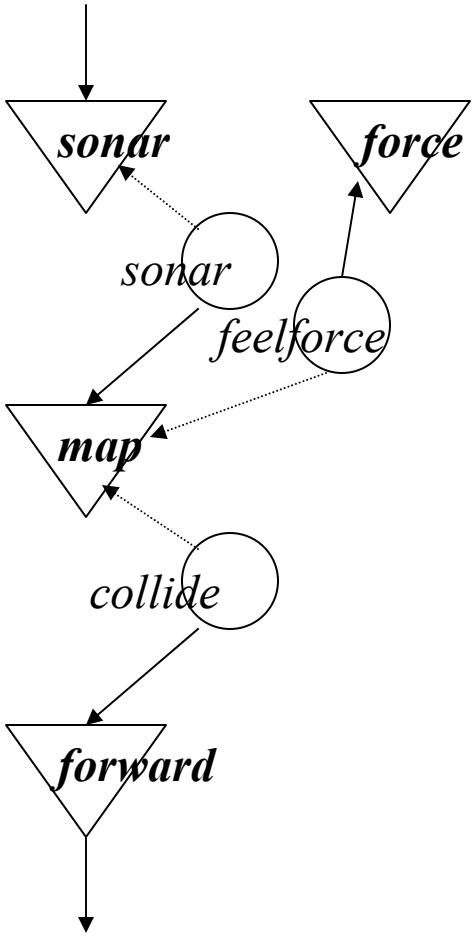
AVOID



When there is a danger of bump, Collide stops robot

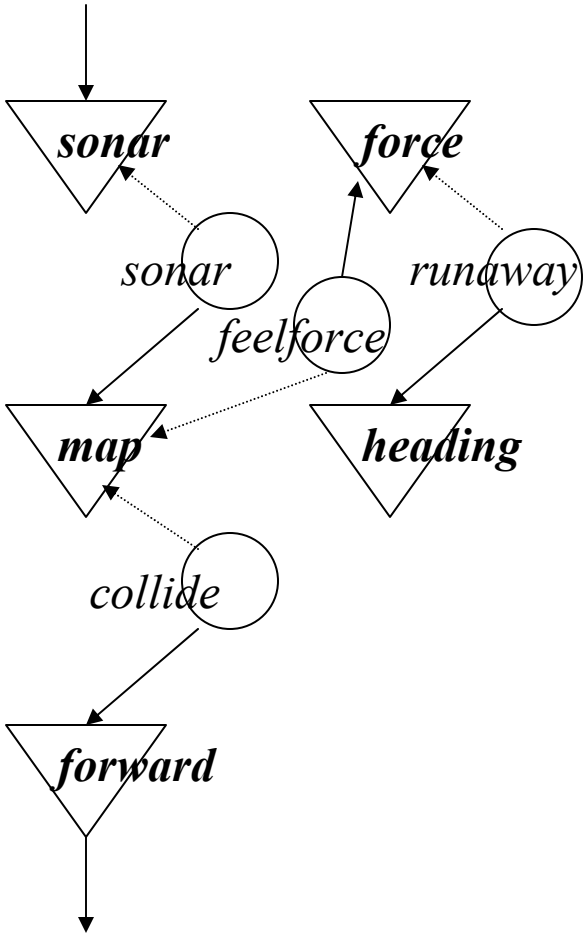
Thus robot goes ahead until it reaches and obstacle, then stops.

AVOID



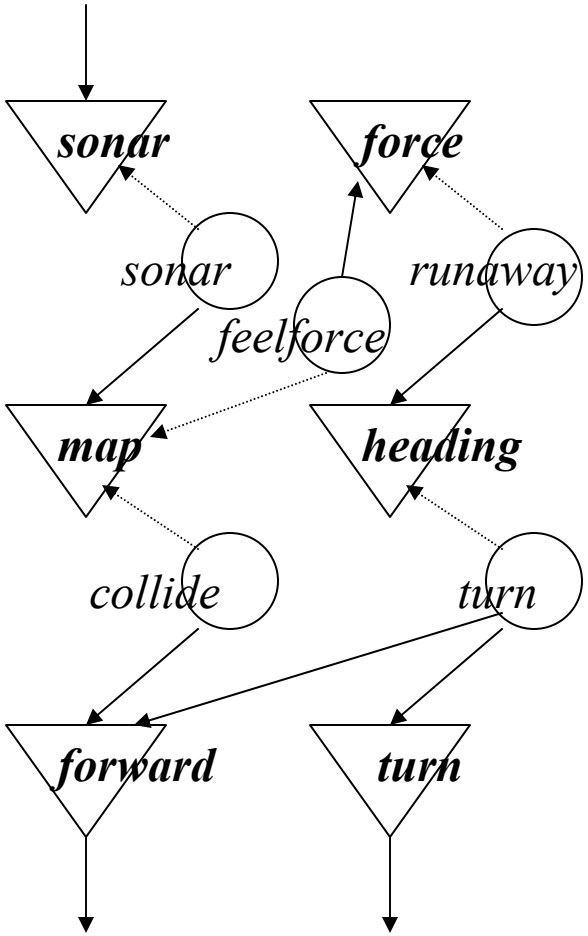
Feelforce evaluates the most dangerous (relative) direction of movement

AVOID



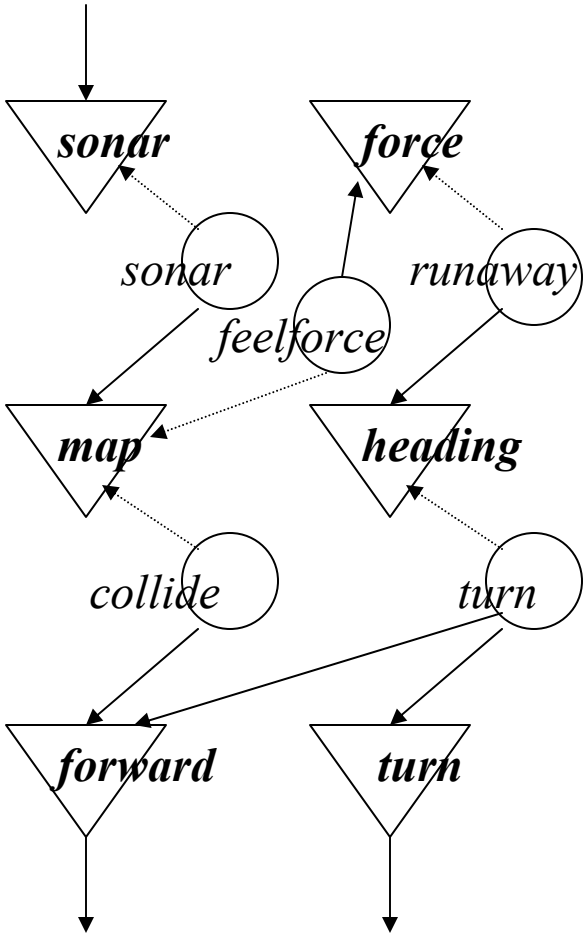
Runaway proposes to turn somewhere else

AVOID



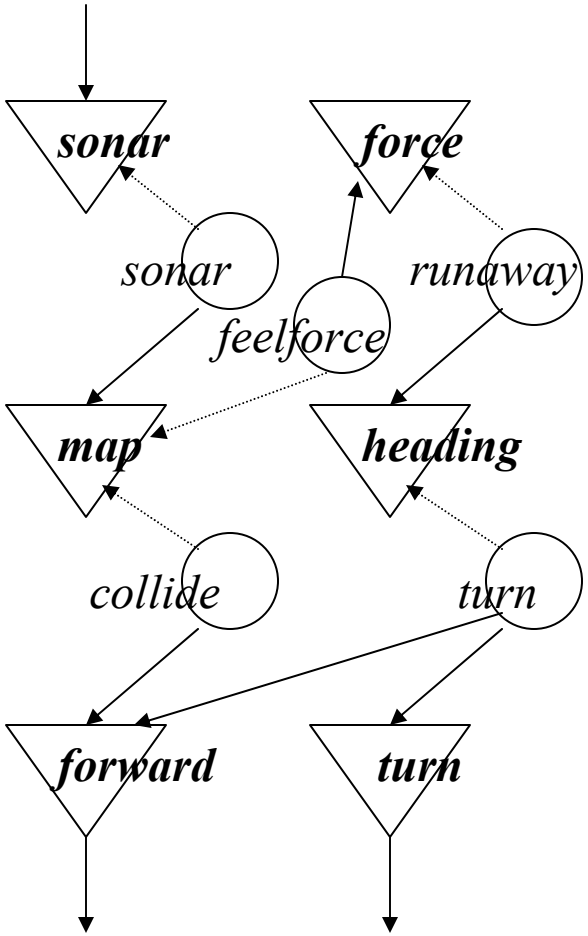
Turn stops and turn to the proposed direction

AVOID



Concern that turning has a feedback to content of map and consequentially to heading. More turn performed, less turn required.

AVOID

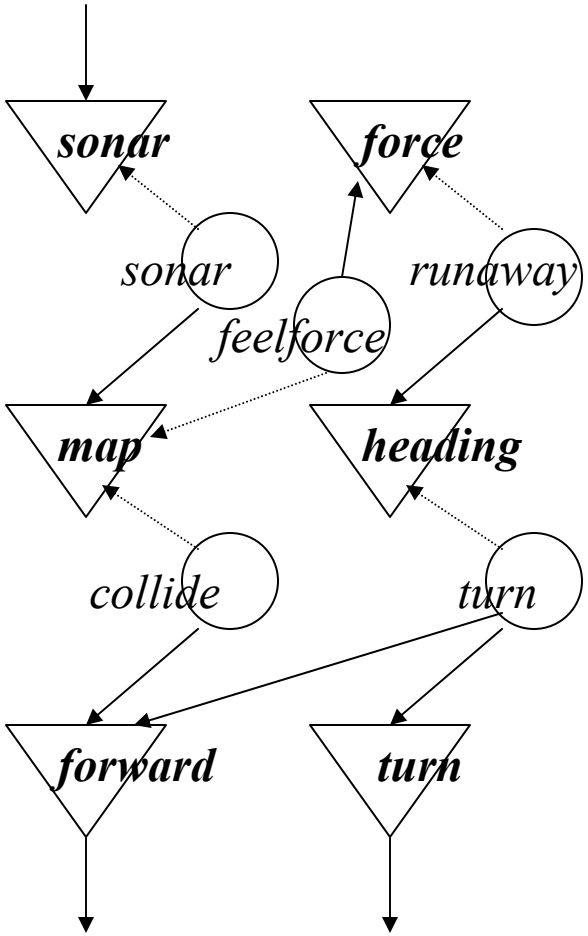


Thus robot go ahead until reaches an obstacle. Then it turns and then follows straightforward. Again turns, again go ahead, repeatedly.

It easily gets on a cyclic path. However it is enough for AVOID.

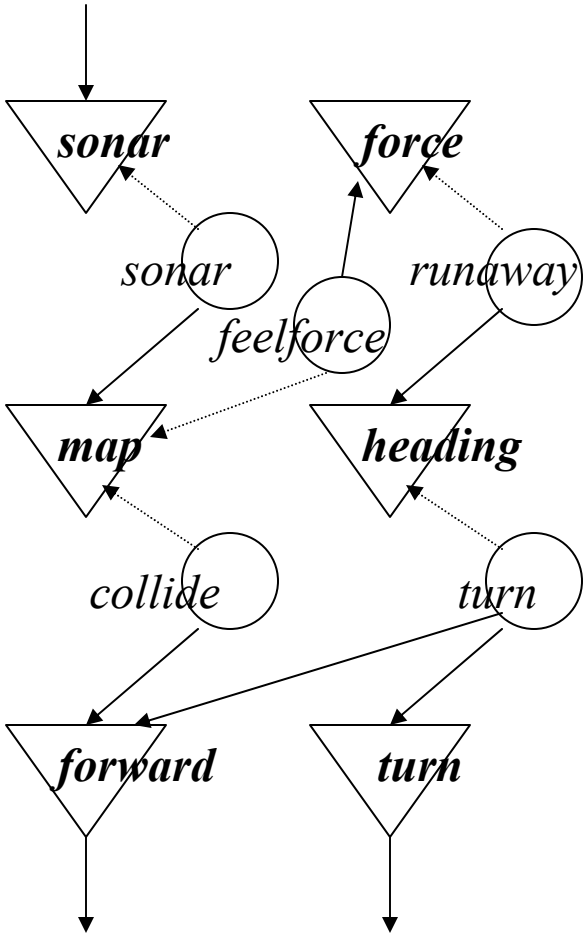
AVOID is finished.

AVOID

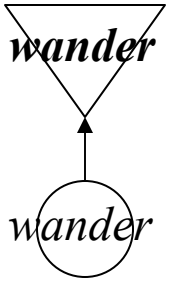


Now it is interesting that by development of avoiding static obstacles we have got also running away from moving objects. (Sometimes we call such situation as emergence)

AVOID



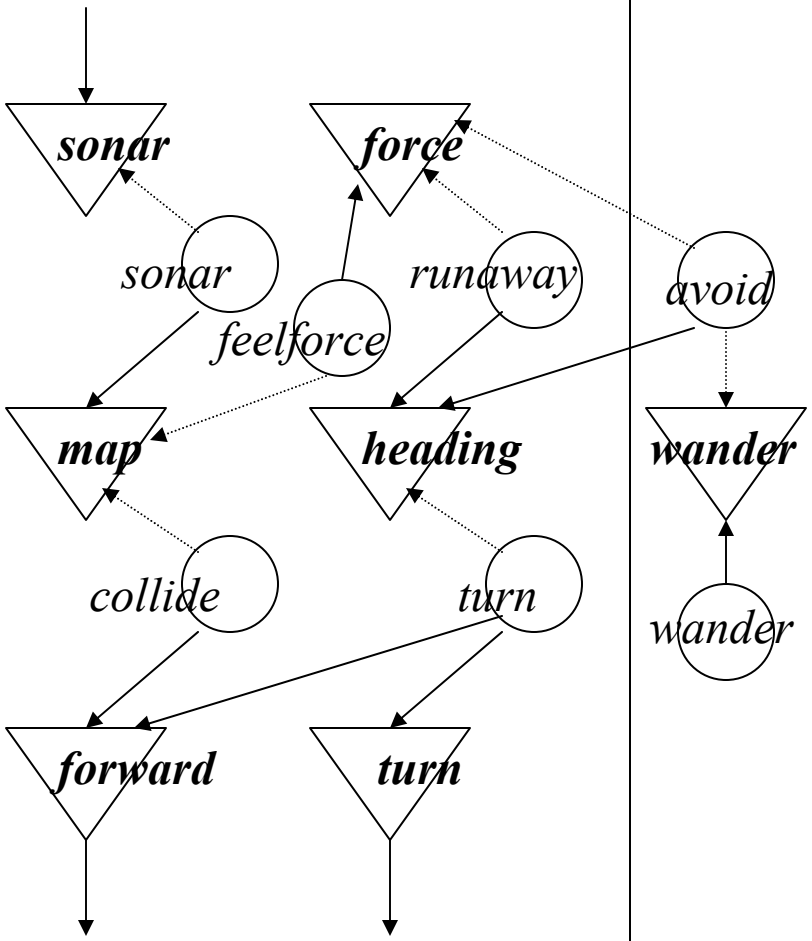
WAN DER



Now we would like to get the robot from the cyclic paths by an occasional random movement

Therefore wander sometimes propose a turn

AVOID



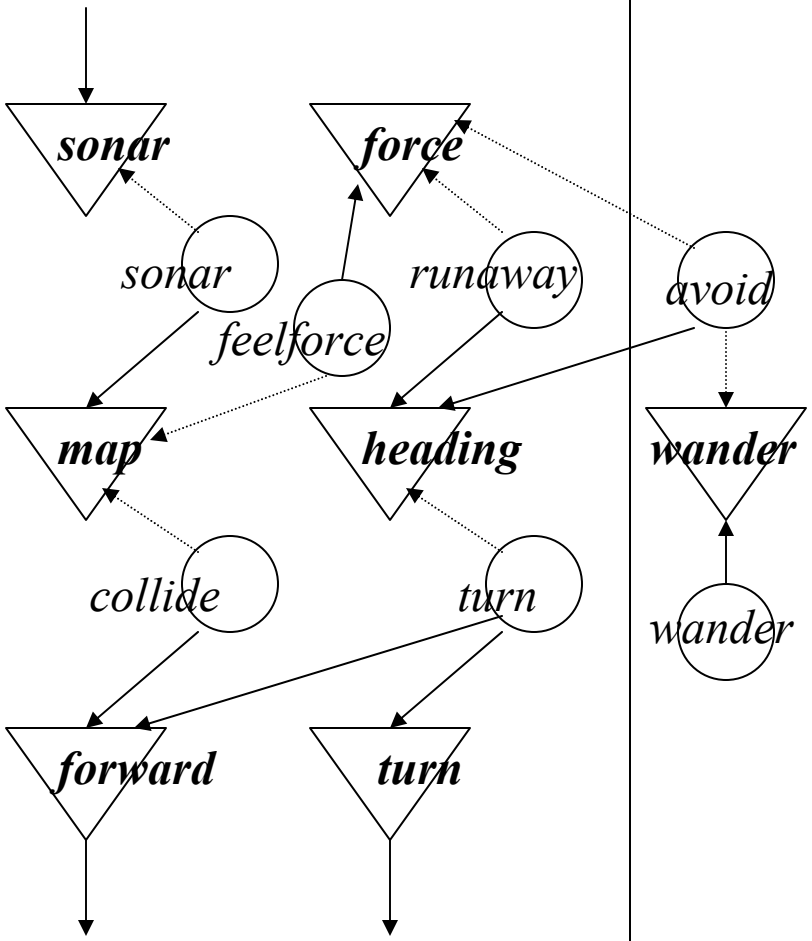
WAN DER

Agent avoid persuade itself that there is no danger of collision and then overwrite heading by the direction proposed by Wander

Thus we employ avoiding obstacles for random wandering, since Agent Turn use the value as written by agent Heading

AVOID

WAN
DER



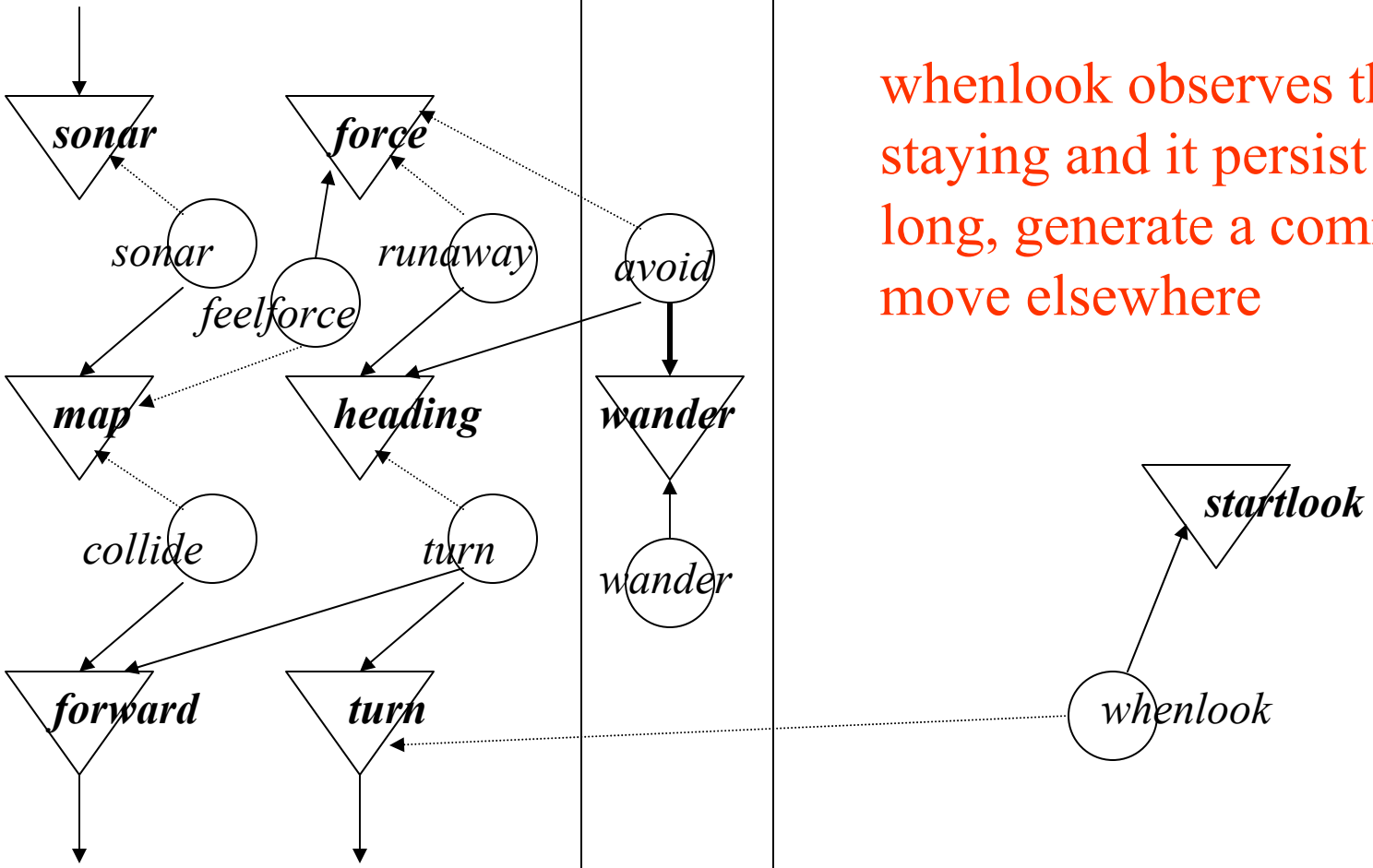
Now robot avoids obstacles but its movement is unpredictable.

But it stays in quite a small area.

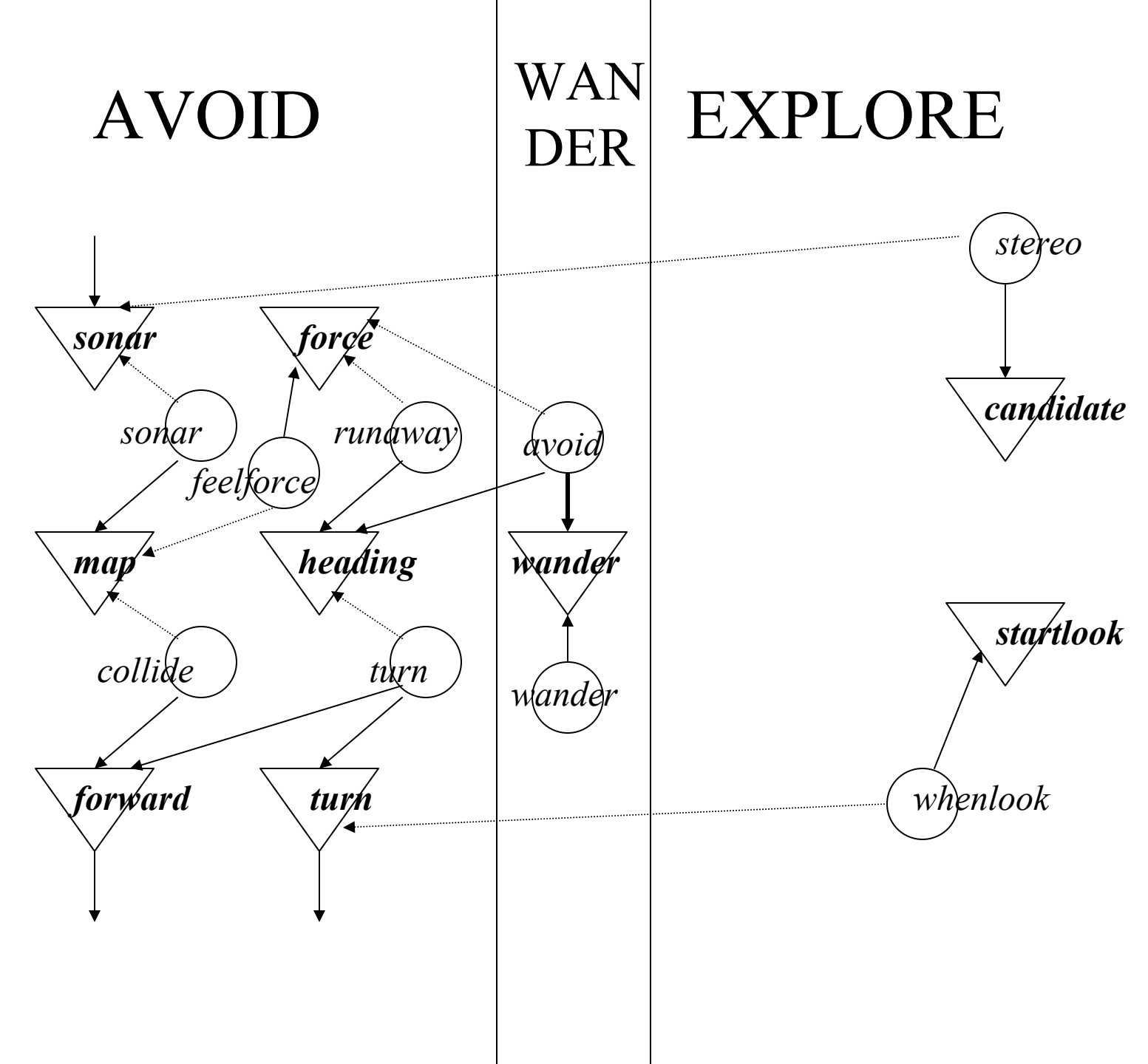
AVOID

WAN DER

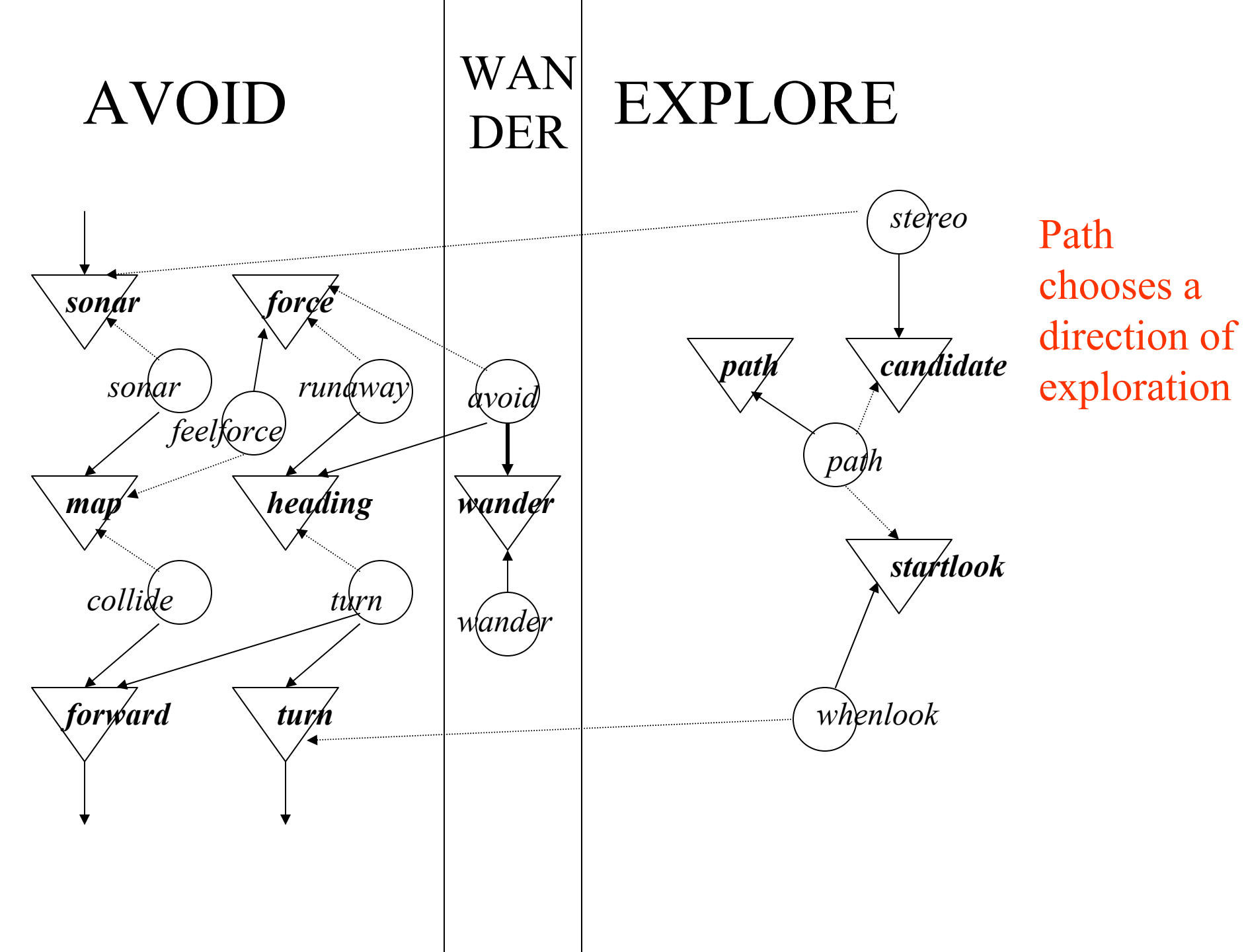
EXPLORE

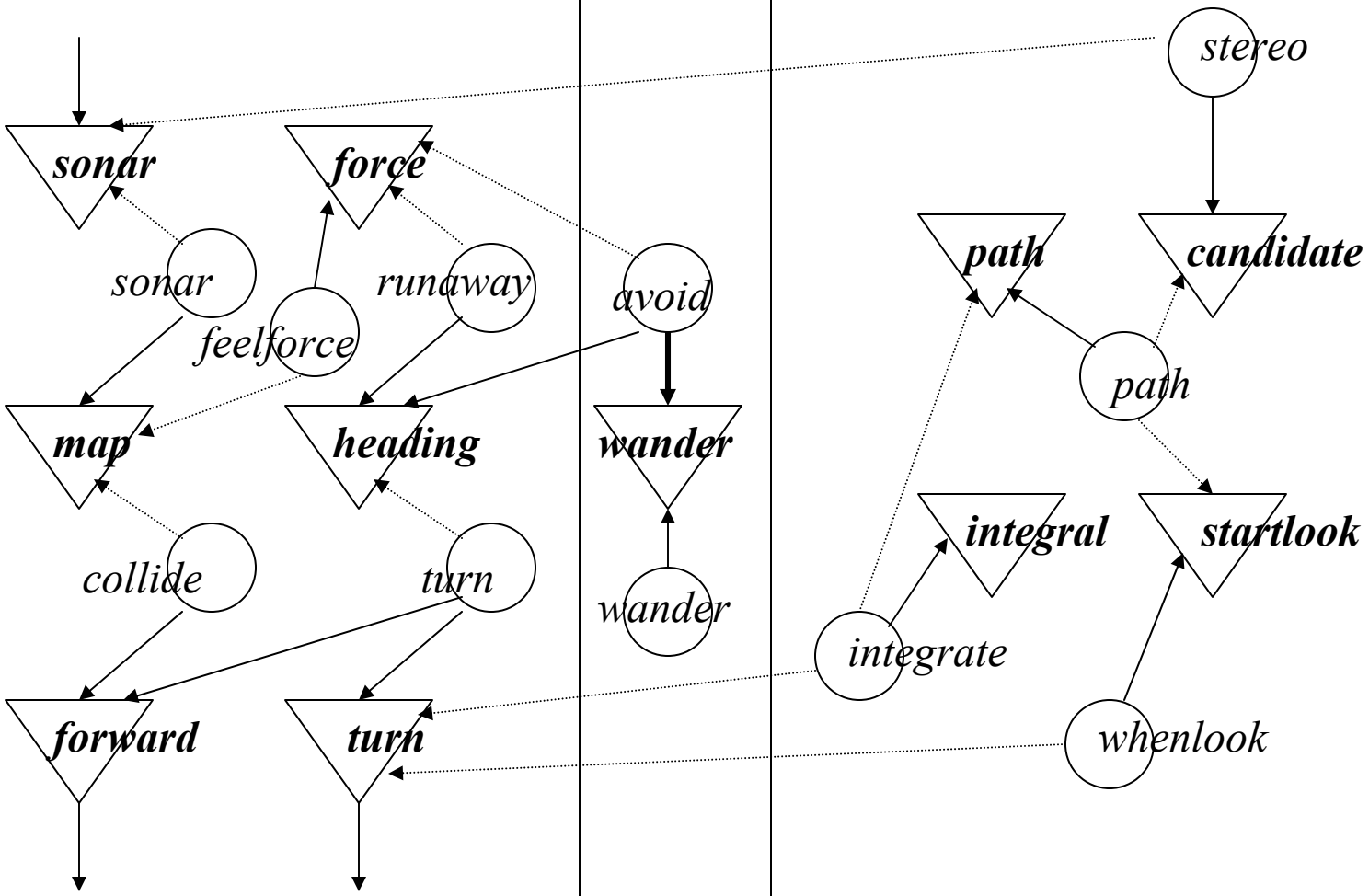


whenlook observes this staying and it persist for too long, generate a command to move elsewhere

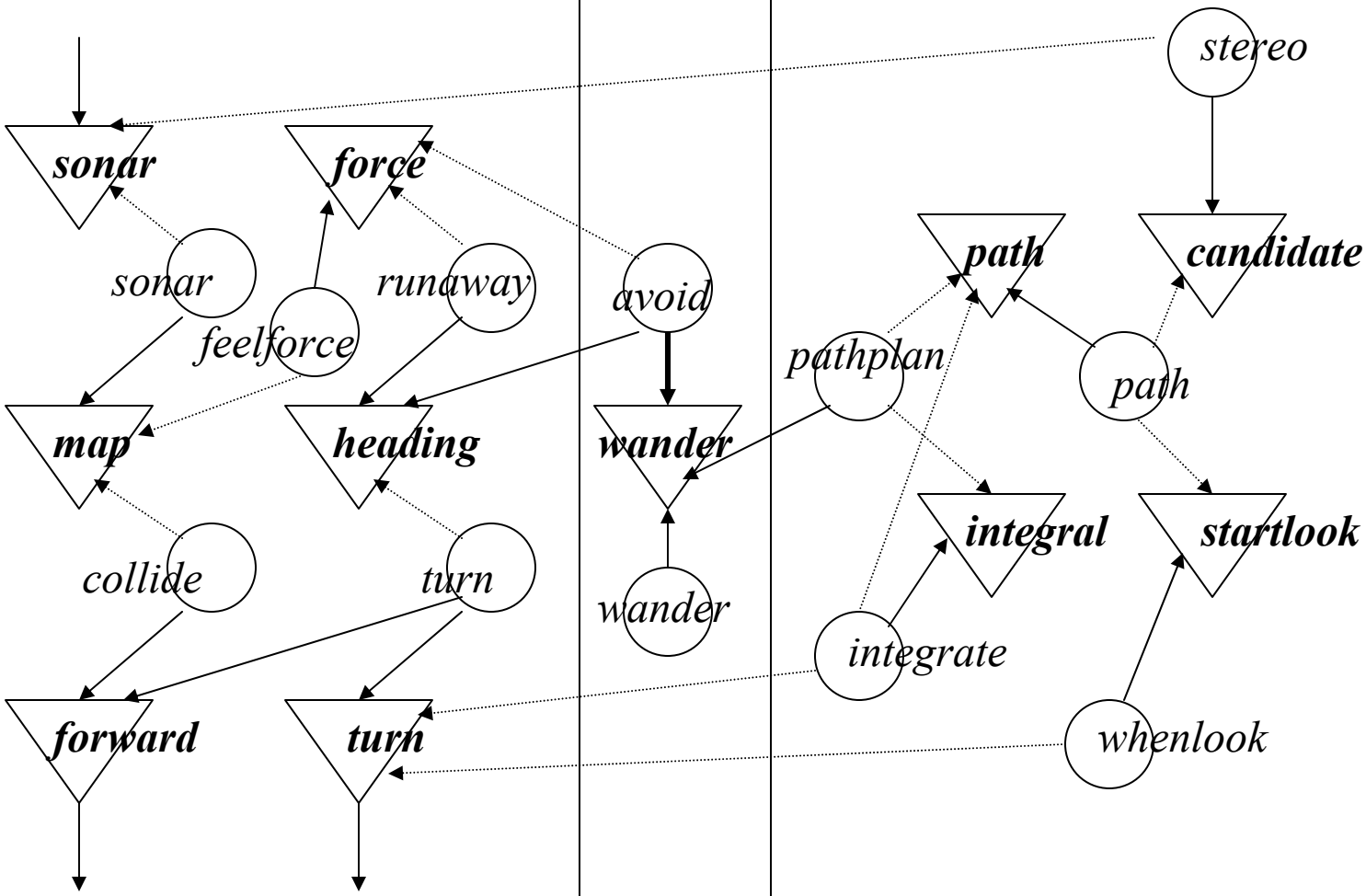


Stereo is looking for possibility of free movement

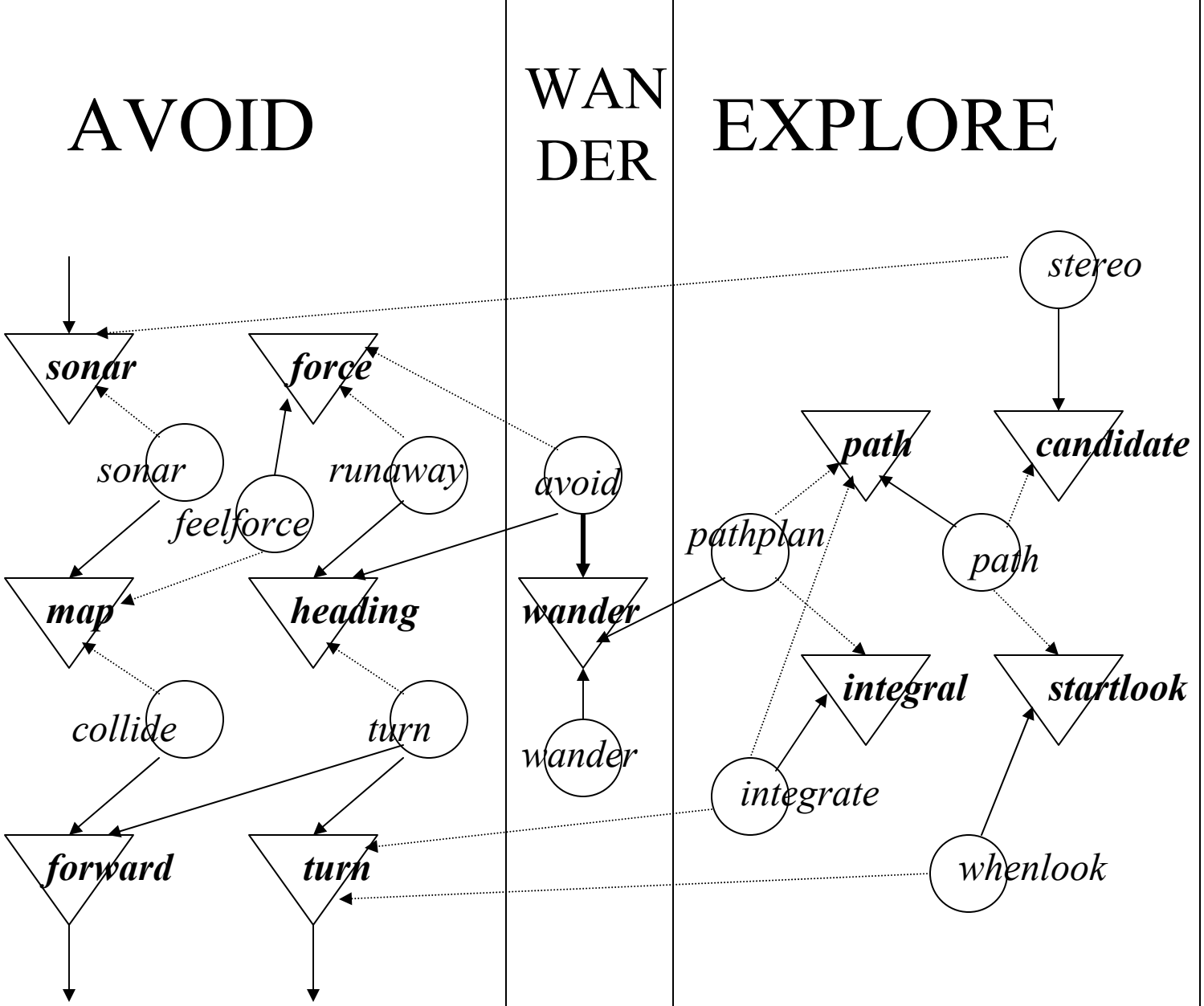

AVOID
WAN DER
EXPLORE
stereo
sonar
force
path
candidate
sonar
runaway
avoid
feelforce
path
map
heading
wander
startlook
collide
turn
wander
forward
turn
whenlook
Path chooses a direction of exploration



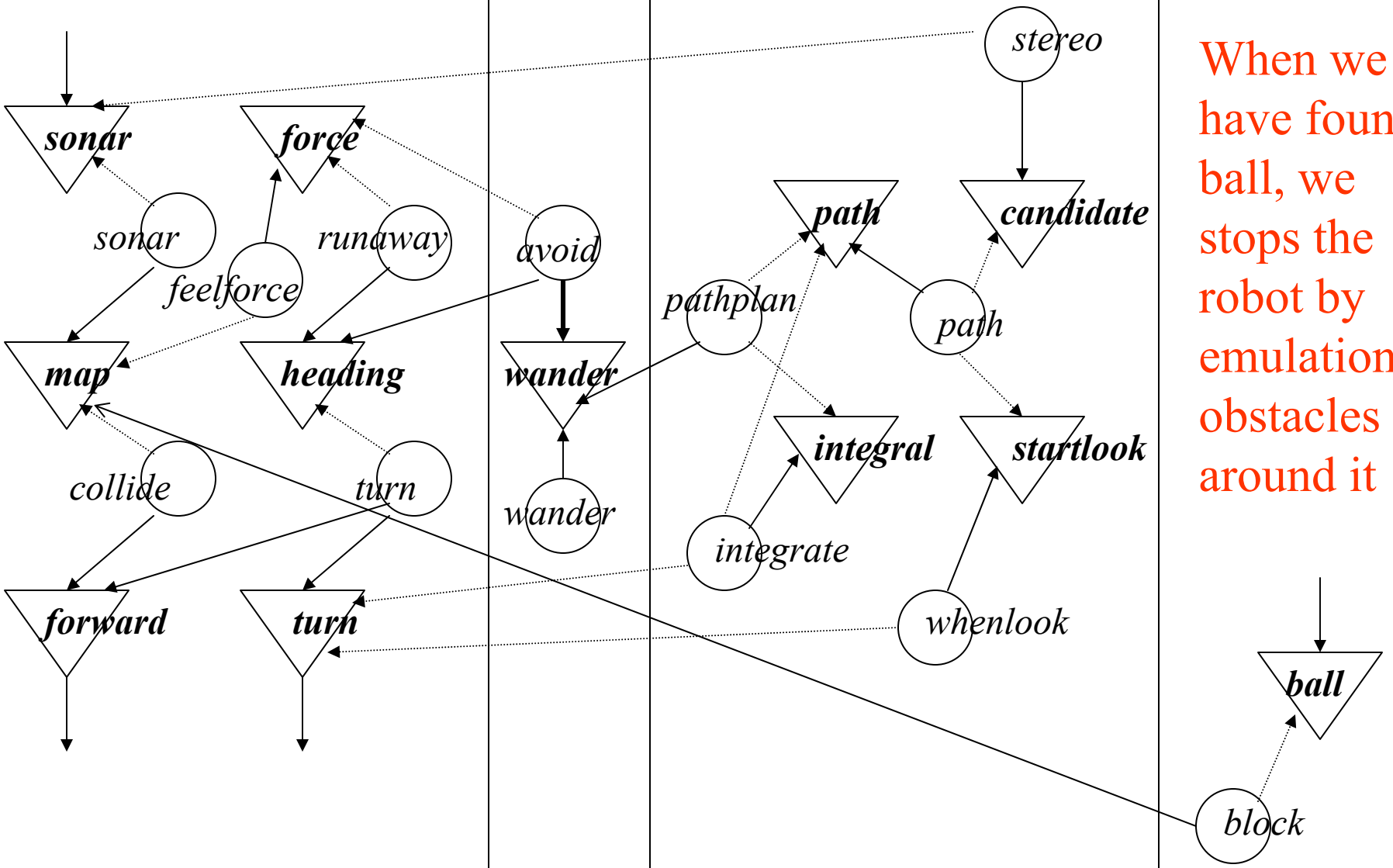
This is only a relative direction, so we need to estimate difference to calculate absolute directions



Pathplan calculate a turn which has to be provide to follow the absolute direction of exploration and propose it as a random movement (which is random at all)



Now we are able to explore the environment



When we have found ball, we stops the robot by emulation obstacles around it

Criticism

- Inaccessibility of internal state
- Disruption of levels of competence
- Data fusion

Alternative solutions

- fine-grained aleternative (Rosenblatt, Payton)
- Behavioral systems (Arkin)

Thank you for your attention !

RNDr. Andrej Lúčny
MicroStep-MIS & DAI FMFI UK
andy@microstep-mis.com
http://www.microstep-mis.com/~andy